IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Kenneth G. Ricks
Serial No.:	09/728,407
Filed:	December 1, 2000
For:	System And Method For Creating Architectural Descriptions Of Real Time Simulations
Group Art Unit:	2123
Examiner:	Thomas H. Stevens
Attorney's Docket No.:	MFS-31524-1
Customer No.:	_____

DECLARATION OF KENNETH G. RICKS TRAVERSING REJECTIONS

UNDER 37 C.F.R. 1.132

1. My name is Kenneth G. Ricks. I am the inventor of the "System And Method For Creating Architectural Descriptions Of Real Time Simulations" described and claimed in the pending U.S. Patent application identified above. Attached to this Declaration as Exhibit 1 is a copy of my resume, which accurately describes my experience in the related fields of real-time multiprocessor architectures, real-time simulations, real-time software scheduling, and high-level software design.

2. I have reviewed the Office Action issued by the U.S. Patent and Trademark Office ("PTO") in this patent application on April 5, 2004. I have also read the prior art relied on by the Patent Examiner in rejecting the claims of our patent application.

3. The invention that is the subject matter of each of the claims in our patent application has solved a problem encountered by those skilled in the art of software simulation: the problem of designing and specifying real-time, hardware-in-the-loop simulations.

4. SADL was created to address the particular problem of designing and specifying real-time, hardware-in-the-loop simulations. These simulations are characterized as having many different software components and hardware components that must cooperatively execute within strict timing requirements in order to perform the desired task. The job of constructing such a simulation is daunting, and without the aid of a design tool, it can often get too overwhelming for even experienced simulationists.

5 There are many tools, languages, and environments that have been created specifically for software intensive systems within some particular problem domain. An exhaustive list of such tools, languages, and environments is beyond the scope of this summary. SADL incorporates some of the features of existing tools and applies these features specifically to the domain of real-time, hardware-in-the-loop simulation design and specification. In addition to this new application of existing features, SADL creates several unique features and applies them to the domain of real-time, hardware-in-the-loop simulation design and specification. SADL, as a whole, is a completely unique design and specification language targeted to solve a very narrow problem domain that existing tools, languages, and environments did not address.

6. Working within the context of the specification of software-intensive systems, there are some aspects of SADL that are adopted from fundamental knowledge of the field. For example, the use of task graphs to graphically describe the overall system is common. Task graphs having nodes representing software components and arcs between the nodes representing the communications among the software components

is common as well as are the execution characteristics of periodic and aperiodic software processes.

Unique Features of SADL

7. The unique features of SADL include the types of components that can be used in the task graph. For example, hardware-in-the-loop components can be part of the graphical representation of the system. Wells, and other prior art do not teach high level software simulation of systems having hardware-in-the-loop components. While Wells references "firmware," that term refers to permanently embedded software, such as read only memory, and does not teach hardware-in-the-loop components.

8. Also, the continuous process type of software components is unique to SADL. The continuous process has execution characteristics that distinguish it from the types of software components found in the field. Continuous processes exist for the duration of the simulation and cannot be classified as periodic or aperiodic. The Wells prior art reference does not teach continuous process simulation. Moreover, the principal author of Wells has stated that continuous process simulation is a significant difference between SADL and the simulations taught in Wells.

8. Similarly, the types of arcs representing the communications among the system components include unique features. Data-only communications and synchronization-only communications are common in the field. SADL includes these components, but also includes a synchronization-with-data communication that is unique. This communication performs a synchronization between the communicating components and passes data from one component to another.

9. Another unique feature of SADL is that all these arcs can connect any type of system component, including hardware-in-the-loop components. Other description languages refer only to communications among software components. In SADL, all communications have been generalized to refer to possible communications between any two graphical components which represent real system components.

10. Further, SADL uses synchronization characteristics of the communications that are unique. Each communication involving synchronization has a release time and a frequency characteristic associated with it. The release time describes at what point during the execution of the parent component the synchronization mechanism is actually transferred to the child component. The frequency describes at what rate the synchronization mechanism is actually sent to the child. Neither of these characteristics is used in other description languages. But, each represents a critical behavioral element of the system.

Conclusion

11. As a whole SADL represents a unique language that is specifically designed for real-time, hardware-in-the-loop simulation design and specification. These simulations can represent months or years of work by multi-disciplinary teams of engineers and can consist of hundreds of software and hardware-in-the-loop components. Without the capability to systematically organize, design, and specify the simulation, the job of constructing such a simulation can be nearly impossible. The capabilities of SADL, taken as a whole, represent a unique solution to this specific

problem that was not addressed by existing tools, languages, or software environments.

12. I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Kenneth G. Ricks

Kenneth G. Ricks

10/04/2004

Date